Computing and Problem-Solving with Pascal

T. Ray Nanney

COMPUTING AND PROBLEM-SOLVING WITH PASCAL

T. RAY NANNEY

Furman University
Greenville, South Carolina

Prentice-Hall, Inc.
Englewood Cliffs, New Jersey 07632

Library of Congress Cataloging in Publication Data

Nanney, T. Ray
Computing and problem-solving with Pascal.
Includes index.
1. PASCAL (Computer program language) 2. Electronic digital computers–Programming. I. Title.
QA76.73.P2N36 1985 001.64'24 84-2071
ISBN 0-13-164799-7

Cover design: Jeannette Jacobs
Manufacturing buyer: Gordon Osbourne

© 1985 by Prentice-Hall, Inc., Englewood Cliffs, New Jersey 07632

All rights reserved. No part of this book
may be reproduced, in any form or by any means,
without permission in writing from the publisher.

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-164799-7 01

Prentice-Hall International, Inc., *London*
Prentice-Hall of Australia Pty. Limited, *Sydney*
Editora Prentice-Hall do Brasil, Ltda., *Rio de Janeiro*
Prentice-Hall Canada Inc., *Toronto*
Prentice-Hall of India Private Limited, *New Delhi*
Prentice-Hall of Japan, Inc., *Tokyo*
Prentice-Hall of Southeast Asia Pte. Ltd., *Singapore*
Whitehall Books Limited, *Wellington, New Zealand*

PREFACE

The purpose of this book is to provide a thorough introduction to computer programming for students who have little background in computing. The major features of the computer language Pascal are covered in detail. Throughout the book a problem-solving approach is emphasized to provide insight into the thinking process associated with writing computer programs. The concepts are presented in a nonmathematical manner, although a few examples require a modest knowledge of high school algebra. No previous knowledge of computers, programming, or other specialized skills is required.

The book should be useful to all college students–regardless of academic major–who wish (or need) to learn to write computer programs using Pascal. It is particularly appropriate for liberal arts courses in which students from many academic disciplines are enrolled. It is also suitable for self-study.

The first five chapters (taken with minor modifications from my earlier text, *Computing: A Problem-Solving Approach with FORTRAN 77*) provide a background for those who have no prior knowledge of computing. Students who have studied another computer language may be able to start with Chapter 6, Pascal Data Types, but most will profit from a study of the introductory chapters. In my classes, the chapter on LOGO turtle graphics is particularly helpful because it introduces quickly many important programming concepts and problem-solving techniques. If a LOGO interpreter is available, it should be used for writing programs and mastering the concepts.

In my beginning computing course, the approaches used in this book have been exceptionally successful in making programming comprehensible to students from a large number of academic disciplines. Nonscience students have performed as well as science students. Nearly all of them report having fun writing programs, and some claim that the course improved their ability to think logically and critically. After completion of the course, many of these students have continued to use a computer to solve problems in their own academic major. These observations make me confident that the techniques used in this book can be helpful to anyone who wants to learn to write computer programs in Pascal.

T. RAY NANNEY

此为试读,需要完整PDF请访问: www.ertongbook.com

CONTENTS

Chapter *Page*

1 COMPUTERS: AN OVERVIEW 1

1.1 Computers in Society 1
1.2 The Speed of the Computer 2
1.3 The Decreasing Cost of Computers 4
1.4 The Computer as Problem Solver 5
1.5 High-Level View of Computer Structure 6
1.5.1 Some Components of the Central Processing Unit 8
1.5.2 The Input/Output Processor 10
1.6 Microcomputers, Personal Computers, and Minicomputers 11
1.7 The Reliability of Computers 12
1.8 Computer Science 14

2 COMMUNICATION, INFORMATION, AND LANGUAGE 18

2.1 A Model of Communication 18
2.2 Information and Communication 19
2.3 Features of Natural Languages 21
2.3.1 Complexity of Language 21
2.3.2 Redundancy 22
2.3.3 Ambiguity 23
2.3.4 Syntax, Semantics, and Logical Validity 24
2.4 Computer Languages 25
2.4.1 Machine and Assembly Language 26
2.4.2 High-Level Languages 27
2.4.3 The Need for Higher-Level Languages 28
2.4.4 Problem-Solving Domains–Various High-Level Languages 28

Chapter *Page*

3 **ALGORITHMS** **33**

3.1 Processes 33
3.2 Effective Procedures 35
3.3 Algorithms 37
3.4 Computability Theory 38

4 **SOME PROGRAMMING CONCEPTS** **40**

4.1 Concept of a Computer Program 41
4.2 Display Turtles 41
4.3 Definition Mode 44
4.4 Parameters 45
4.5 Names and Values 46
4.6 Bugs and Debugging 47
4.7 Subprocedures 48
4.8 Programming Examples 50
4.9 Recursive Procedures 52
4.10 Planning and Top-Down Programming 55

5 **COMPLEXITY AND COMPUTER PROGRAMMING** **63**

5.1 Human Limitations 64
5.2 Structured Programming 65
5.2.1 The Size of Complex Programs 65
5.2.2 Testing Complex Programs 66
5.2.3 Does Structured Programming Work? 66
5.2.4 Rules for Structured Programming 67
5.3 Top-Down Development of Programs 73
5.3.1 Pseudocode 73
5.3.2 Examples of the Use of Pseudocode 74
5.3.3 Flowchart Technique 81

6 **PASCAL DATA TYPES** **86**

6.1 Input/Output Strategy for Program Development 87
6.2 The Concept of Data Type 94
6.3 Pascal Scalar Data Types 96
6.3.1 Type INTEGER 97
6.3.2 Type REAL 98

Chapter **Page**

6.3.3 Type CHAR 99
6.3.4 Type BOOLEAN 101
6.3.5 Enumerated Types 101
6.3.6 Scalar Subrange Types 102
6.4 Other Pascal Data Types 103

7 **OVERVIEW OF PASCAL** **109**

7.1 High-Level Organization of a Pascal Program 110
7.2 The VAR Declaration Part 113
7.3 The CONST Definition Part 115
7.4 Names in Pascal 116
7.5 Assignment Statements 118
7.6 Compound Statements 121
7.7 Control Statements 122
7.8 Input/Output Statements 124
7.8.1 READ and READLN Statements 125
7.8.2 The EOF Function 128
7.8.3 Simple WRITE and WRITELN Statements 129
7.9 Predefined Procedures and Functions 131
7.10 Sample Programs 132

8 **ARITHMETIC** **151**

8.1 Assignment Statements 151
8.2 Precedence of Operations 155
8.3 More About Output 158
8.3.1 Expressions in WRITELN 158
8.3.2 Carriage Control 162
8.3.3 Printing Reports 162
8.4 Standard Functions 168
8.4.1 Characteristics of Standard Functions 168
8.4.2 Examples Using Standard Functions 171
8.5 Arithmetic Errors 174
8.5.1 Integer Overflow 175
8.5.2 Overflow and Underflow for Real Numbers 177
8.5.3 Addition and Subtraction of Real Numbers 179
8.5.4 Inexact Representation of Real Numbers 181
8.6 Rounding 185
8.7 Other Errors 187

Chapter		*Page*
9	**DECISIONS**	**199**
9.1	IF . . . THEN . . . ELSE Statement	199
9.2	BOOLEAN Data Type–Variables, Expressions, and Statements	213
9.3	ELSE IF Form of the IF Statement	223
9.4	The CASE Statement	234
10	**LOOPS**	**249**
10.1	The Structure of Loops	249
10.2	The REPEAT Statement	251
10.3	The FOR Statement	253
10.3.1	The Syntax of the FOR Statement	255
10.3.2	The Semantics of the FOR Statement	257
10.3.3	A Sample Program	262
11	**CHARACTER MANIPULATION**	**276**
11.1	Character Sets	276
11.2	Standard Functions for Characters	277
11.3	Declaration of Character Variables	279
11.4	Assignment Statements Involving Characters	282
11.5	Comparison of Variables of Type CHAR	283
11.6	Input/Output of Characters	286
11.7	Programming Examples	291
11.7.1	Printing the Contents of a File	291
11.7.2	Palindromes	293
11.7.3	Distribution of Final Class Grades	297
12	**ONE-DIMENSIONAL ARRAYS**	**311**
12.1	Declaration of Arrays	312
12.2	Assignment Statements Using Arrays	315
12.3	Comparisons Involving Arrays	316
12.4	Programming Examples	318
12.4.1	Bank Checking Account Balance	318
12.4.2	Frequency Distribution	320
12.4.3	Prime Numbers, Sieve of Eratosthenes	327
12.4.4	Standard Deviation	330
12.4.5	Frequency of Occurrence of Characters	333

Chapter *Page*

13 **PROCEDURES AND FUNCTIONS** **348**

13.1 An Introductory Example 349
13.2 Declaring Procedures 353
13.2.1 Local Variables 353
13.2.2 Formal Parameters 356
13.3 Using Procedures 360
13.4 Functions 365
13.5 Block Structure 367
13.6 Programming Examples 370
13.6.1 Sorting 370
13.6.2 Table Lookup: Payroll Calculation 379
13.6.3 Manipulation of Strings 389
13.7 Recursive Subprograms 401

14 **MULTIDIMENSIONAL ARRAYS** **417**

14.1 Descriptions of Two- and Three-Dimensional Arrays 417
14.2 Declaration of Multidimensional Arrays 423
14.3 Input/Output of Two- and Three-Dimensional Arrays 425
14.4 Programming Examples 427
14.4.1 Student and Class Averages 427
14.4.2 Frequency Distributions 435
14.4.3 Questionnaires 441
14.4.4 Translating Morse Code 453
14.4.5 Eight Queens Problem 465

15 **RECORDS** **488**

15.1 Declaration of Records 489
15.2 Operations with Records 493
15.3 The WITH Statement 498
15.4 Variant Records 503
15.5 Programming Examples 507
15.5.1 Student Records 507
15.5.2 Class Rank of Students 510
15.5.3 Simulation of Bank Teller Machines 515

Chapter *Page*

16 OTHER PASCAL DATA TYPES 534

16.1 Files 535
16.1.1 Definition and Declaration of Files 535
16.1.2 Input/Output of Files 537
16.1.3 Textfiles 541
16.1.4 The Standard Files INPUT and OUTPUT 542
16.1.5 Programming Examples 543
16.2 Sets 548
16.2.1 Declaration and Construction of Sets 549
16.2.2 Set Operators 553
16.2.3 Programming Examples 555
16.3 Pointers 560
16.3.1 Dynamic Data Structures 562
16.3.2 Declaration of Pointer Variables 565
16.3.3 Operations with Pointer Variables 567
16.3.4 Programming Examples 571

Appendixes

I Programming Errors and Debugging 592

I.1 Errors in Control Commands 593
I.2 Syntax Errors 593
I.3 Logic Errors 594

II Encoding Schemes for Character Sets 599

III Syntax Diagrams 601

Answers to Review Quizzes 609

Answers to Selected Exercises 611

Index 639

此为试读，需要完整PDF请访问： www.ertongbook.com

LIST OF FIGURES

Figure		*Page*
1.1	Some Levels of Knowledge for Hardware and Software	6
1.2	Major Components of a Computer	7
1.3	Functional Description of a Small Time-Sharing Computer	10
1.4	A Structure of Computer Science	14
2.1	General Model of Communication	18
3.1	Some Flowchart Symbols	35
3.2	Part of a Flowchart for Making French Bread	36
3.3	Algorithm for Determining Whether or Not an Integer Is a Prime	38
4.1	CRT Screen for Display Turtles	42
4.2	Different Procedures for Drawing a Square	51
5.1	The Sequence Structure	69
5.2	The IF Structure	69
5.3	Use of the IF Structure	70
5.4	The WHILE Structure	71
5.5	Use of the WHILE Structure	72
5.6	Program in Flowchart Form: Class Average for a Quiz	82
5.7	Flowchart of a Golf Swing	83
6.1	Bank Checking Account Summary	89
6.2	Sample Bank Statement	92
6.3	Hierarchy of Pascal Data Types	96
7.1	Greatest Common Divisor–Euclidian Algorithm	136
7.2	Manhattan Island Problem	139
7.3	Prime Numbers	143
7.4	Prime Numbers–Limit Method	146

Figure *Page*

8.1 Compute Simple Payroll with Overtime 167
8.2 Manhattan Island Problem 169
8.3 Right Triangles 173
8.4 Roots of the Quadratic Equation 174
8.5 Calculation of Pi 180
8.6 Comparison of Summation Techniques 182

9.1 Structure of the IF Statement 200
9.2 Calculation of the Roots of the Quadratic Equation 206
9.3 Scoring of Bowling 214
9.4 Scoring of Bowling: Illustration of Boolean Variables 225
9.5 Quiz Grades: Illustration of ELSE IF Statement 229
9.6 Quiz Grades: Illustration of Nested IF Statements 230
9.7 Scoring of Bowling: Illustration of ELSE IF Construct 232

10.1 Comparison of the WHILE and REPEAT Structures 252
10.2 Grade Statistics 254
10.3 Comparison of the FOR and WHILE Structures 258
10.4 Three-Digit Numbers Equal to the Sum of the Cubes of Their Digits 265

11.1 Listing of a File 292
11.2 Palindromes 298
11.3 Distribution of Final Class Grades 300
11.4 Distribution of Final Class Grades Using Arrays 303

12.1 Checking Account Summary 320
12.2 Frequency Distribution of Grades 323
12.3 Frequency Distribution of Grades: 10-Point Intervals 326
12.4 Prime Numbers: Sieve of Eratosthenes 329
12.5 Standard Deviation–Method 1 334
12.6 Standard Deviation–Method 2 335
12.7 Frequency of Occurrence of Characters in a File 338

13.1 Print the Contents of a File: Illustration of Procedures and Global Variables 350
13.2 Print the Contents of a File: Illustration of Procedures Having Local Variables 354
13.3 Block Structure for a Standard Deviation Program–1 368
13.4 Block Structure for a Standard Deviation Program–2 369
13.5 Sorting: Exchange Method 374
13.6 Sorting: Improved Exchange Method 376
13.7 Alphabetic Sorting: Improved Exchange Method 379
13.8 Simple Payroll Reports Using Binary Search 387
13.9 Manipulation of Strings 397
13.10 Towers of Hanoi 404

Figure *Page*

14.1 Compute Student Averages and Quiz Averages 433
14.2 Distribution of Grades Using Procedures 439
14.3 Distribution of Grades by Sex and Class 442
14.4 Sample Questionnaire 446
14.5 Questionnaire Summary 454
14.6 Translate Morse Code to English 466
14.7 Eight Queens Problem 475

15.1 Illustration of Operations with Records 502
15.2 *Student Records: Illustration of Variant Records* 511
15.3 Average Grades and Sort by Average 515
15.4 Simulation of Bank Teller Machine 521

16.1 Construct a File 546
16.2 Student Averages: Illustration of Files 549
16.3 Count the Number of Vowels in a Message 557
16.4 Ignore Duplicates of a Letter in a Word 558
16.5 Demonstration of Set Construction and Intersection Operation 561
16.6 Linked List of Customer Names 574
16.7 Linked List of Names in Alphabetical Order 578
16.8 Build a Binary Tree and List Its Contents in Infix Order 582

LIST OF TABLES

Table		*Page*
1.1	Number of Computers in the United States	3
1.2	Ownership of General-Purpose Conventional Computers within the United States, 1976	3
1.3	Some Input/Output Devices	7
1.4	Comparison of the ENIAC and Fairchild F8 Computers	11
1.5	Comparison of an IBM Personal Computer and an IBM 360-40	12
2.1	Frequency of Occurrence of Characters of the English Alphabet	22
2.2	Corresponding Assembly- and Machine-Language Instructions: C = A + B	27
2.3	Some High-Level Computer Languages	29
4.1	Simple Instructions for Display Turtle	42
5.1	Bugs Observed in Information Bank Programs for the *New York Times*	67
7.1	Reserved Words	117
7.2	Standard Identifiers	118
8.1	Some Pascal Standard Functions	171
9.1	Definition of the Boolean Operators	217

chapter 1

COMPUTERS: AN OVERVIEW

There is hardly an area of human endeavor that has not been affected by computers. Banks use computers to print your bank statements; utility companies compute your usage of their services and a computer prints your bill; computers are used to control the inventory in department stores and supermarkets; airline personnel check your reservations using a computer. They are everywhere–in business, industry, government, education, sciences, and all present-day technology. This widespread use of computers in our society is due primarily to the generality of tasks that computers can perform, the tremendous speed of computer operations, and the decrease in the cost of computers.

1.1 COMPUTERS IN SOCIETY

In 1974, I heard an IBM executive refer to the rapid spread of computer usage as "the insidious revolution." He explained that the general population rarely saw a computer, almost never worked with a computer (computer billing being an exception), and was unaware of the diversity of computer applications and their pervasiveness in society. His claim was that without the public's realization we had become so dependent on computers that society could not have been maintained without them.

In recent years, the public's awareness of computers has changed dramatically, primarily because powerful personal computers are available at a reasonable cost. Most bookstores sell technical books written for the owners of personal computers; numerous books are available to guide the public in selecting a personal computer. Newsstands sell many magazines that deal with particular personal computers. The availability of computers for a significant segment of the population will have an important impact on society.

Let us expand our perspective regarding computer applications by briefly considering the history of modern computing.*

The first large-scale electronic computer, ENIAC, was completed in 1946 and was used by the U.S. Army Ordnance Corps primarily to compute ballistic tables. In the early 1950s all computers were owned by the federal government and were used for census studies, weapons calculations, weapons delivery and control, cryptographic applications, nuclear design, nuclear engineering, inventory, and logistic applications. In 1954, the UNIVAC I computer was delivered to General Electric Company for commercial use, and the computer explosion had begun.

In a 1967 report to the president of the United States,† it was estimated that 80,000 computers would be in use by the end of 1975. This estimate was far too low. Improved computer technology and the resulting reduced costs made the acquisition of a computer attractive to many organizations. By 1976, the number of computers in use was 220,000 (see Table 1.1); the government's share had dropped to only 3.4 percent (see Table 1.2). Most of the computers were owned by the manufacturing industry, miscellaneous businesses, and banking institutions.

Computers have been installed, replaced, and upgraded at such a rapid rate since 1976 that current statistics would be inaccurate after only a few weeks. If you think of all the computers (or computer-dependent devices) that you see in just one day, you can better understand just how omnipresent computers are.

1.2 THE SPEED OF THE COMPUTER

The ability of computers to perform tasks at amazing speeds is well known. Reasonably fast computers can perform more than 10 million elementary operations in a second and simultaneously operate many attached devices. For example, a moderate-sized computer could, in 1 second, do all the following:

1. Calculate the payroll for more than 1000 employees.
2. Print more than 100 checks.
3. Update records for 2000 students.
4. Grade and analyze 3000 exams of 100 questions each.

This speed gives us tremendous power to manipulate information.

There is another aspect of computing power to be considered: the power of com-

*Much of this information was abstracted from Ruth M. Davis, "Evolution of Computers and Computing," *Science, 195* (March 18, 1977), 1096-1102, Tables 1 and 2; and Saul Rosen, "Electronic Computers: A Historical Survey," *Computing Surveys, 1* (March 1969), 7-36. Copyright 1969, Association for Computing Machinery, Inc.; by permission.

†*Computers in Higher Education*, Report of the President's Science Advisory Committee, The White House, Washington, D.C., February 1967, p. 58.

此为试读,需要完整PDF请访问: www.ertongbook.com